Electronic Version
Stylesheet Version v1.1.1

Description

METHOD AND APPARATUS FOR IMPROVING THE MANAGAMENT OF DATA PACKETS

BACKGROUND OF INVENTION

[0001] 1. Field of the Invention

[0002] The invention relates to a network system, and more particularly, to an apparatus and method for improving the management of received data packets.

[0003] 2. Description of the Prior Art

[0004] A network device (e.g., network interfaced card, NIC) receives data packets from a network and transmits data packets from a host system. When a NIC receives a data packet from the network, the NIC will transfer the data packet into a data buffer located in the host system. The host system has a plurality of data buffers with some data buffers being in a "free" state (able to store data packet) and some data buffers being in an "unavailable" state

(unable to store data packet). In order to locate a free data buffer, the NIC refers to descriptors. Descriptors are memory structures, which can be located either on the NIC or the host system, that correspond to a data buffer in the host system; a descriptor contains an own bit and the address and buffer length of a data buffer in the host system. Generally, the number of descriptors is a subset of the number of data buffers in the host system. In other words, there are fewer descriptors than data buffers.

[0005] The descriptors fall into one of two classes: free descriptors and unavailable descriptors. The free descriptors have corresponding data buffers in a free state and the unavailable descriptors have corresponding data buffers in an unavailable state. By referring to a free descriptor, the NIC is able to transfer a data packet into data buffers in a free state. When a data packet is transferred into the data buffers, the NIC marks the own bit of the corresponding descriptor to unavailable. The own bit of an unavailable descriptor is only marked free by the host system when either the host system clears the data buffer corresponding to the host system or when the host system associates the descriptor with a different data buffer that is in a free state.

[0006] The NIC issues an event signal to the host system to indicate what kind of data packet will be transferred into the data buffers. Data packets fall into one of two classes - ok packets and error packets - so naturally the NIC has two signals to represent the packets - an ok signal and an error signal. For example, if the NIC issues an ok signal, then the host system will post the data packet to the protocol stack and associate the descriptor corresponding to the data buffer used to store the data packet with a different data buffer that is in a free state. If the NIC issues an error signal, then the host system will clear the data buffer used to receive the transferred data packet and renew the buffer length of the descriptor corresponding to the data buffer.

[0007] Every signal issued by the NIC is allowed to notify the host system. This, however, makes for inefficiency in the management of the data packets received by the host system. For a string of received ok packets, this is not inefficient because the host system needs to address an ok packet as soon as it is received. However, for a string of received error packets, this is highly inefficient because the host system does not need to address an error packet as soon as it is received. For example, if 4 error packets are received

in a row, then host system will halt its operations and clear the data buffer 4 times.

[0008] A prior art solution counts the number of times an event occurs and notifies the host system via a signal when the number of times an event occurs has reached a certain value. In this way, the prior art solution is able to reduce the number of times the host system interrupts its operations and diverts its resources and power. In the prior art solution, if the number of events counted that triggers the coalesced signal is too high, there is a danger that all the descriptors will be unavailable. With no free descriptors, the NIC will have not be able to obtain the address of a data buffer in a free state in the host system, meaning that the host system will drop the incoming data packets.

[0009] Some may argue that prior art will always have free descriptors available so long the number of events counted that triggers the coalesced event is not set to the total number of descriptors. However, this is not true as an error packet that is received may actually occupy more than one data buffer, which means that more than once descriptor will be marked as unavailable. Each packet includes length information. When the length information of the error packet is erroneous, the length of the error

packet may exceed the length of the data buffer. As a result, more than one data buffer from the host system is needed, which means more than one descriptor will be marked from free to unavailable.

Summary of Invention

[0010] It is therefore one of objectives of the claimed invention to provide an apparatus and related method for improving the management of data packets received from a network by a host system to solve the above-mentioned problem.

[0011] According to the claimed invention, an apparatus for improving the management of received data packets of a host system that comprises a plurality of data buffers and a plurality of descriptors that corresponds to a subset of the plurality of data buffers to manage the received data packets, the apparatus comprising: a receiver for receiving a data packet; a first storage unit for storing the data packet from the receiver; a counter for monitoring the number of descriptors in a first state to produce a count value; a second storage unit for storing a threshold value; a comparator for comparing the count value with the threshold value and producing a comparison signal; and a control logic for issuing a first event to the host system according to the comparison signal.

[0012] According to the claimed invention, a method for improving the management of received data packets of a host system that comprises a plurality of data buffers and a plurality of descriptors that corresponds to a subset of the data buffers to manage the received data packets, the method comprising: receiving a data packet; transferring the data packet into at least one of the data buffers; monitoring an amount of the descriptors in a first state; comparing the amount with a threshold value to generator a comparison signal; and generating a first event to the host system according to the comparison signal to prevent all the descriptors from being in the first state.

[0013] These and other objectives of the claimed invention will no doubt become obvious to those of ordinary skill in the art after reading the following detailed description of the preferred embodiment that is illustrated in the various figures and drawings.

Brief Description of Drawings

[0014] Fig.1 is a block diagram of an apparatus according to the present invention.

[0015] Fig.2 is a flow chart of a method employed by an apparatus according to Fig.1.

Detailed Description

[0016] Fig.1 is a block diagram of an apparatus according to the present invention. In this embodiment, the apparatus 10 comprises a transceiver 20 for receiving data packets, a first storage unit 30 for storing a data packets from the transceiver, a Receive DMA 40 for transferring data packets from the first storage unit 30 into the data buffers 72, a counter 46 for monitoring the number of descriptors in a certain state starting from the initial value, a second storage unit 44 for storing a threshold value that indicates the count value that should cause the apparatus to trigger a first event, a comparator 42 for comparing the count value against the threshold value, and a masking circuit 50 for masking signals from notifying the host system when the masking circuit is on. The apparatus 10 is coupled to a host system 60 via a bus. The host system 60 comprises a memory 70 and CPU 80. The memory 70 comprises data buffers 72 and descriptors 74.

[0017] The comparator 42, second storage unit 44, and counter 46 are not necessarily limited to being positioned within the receive DMA 40, but is only shown so as an example embodiment. Also, the masking circuit 50 is an optional component of the apparatus 10 and should not be taken as a limitation.

[0018] Please refer to Fig 2. Fig.2 is a flow chart of a method employed by an apparatus 10 according to Fig.1. The method comprises the following steps:

[0019] Step 100:Start.

[0020] Step 110:The transceiver 20 receives a data packet from the network and stores the data packet in the FIFO 30.

[0021] Step 120:The receive DMA 40 transfers the data packet from the FIFO 30 to at least one of data buffers 72 of the host system 60.

[0022] Step 130: If the data packet is an ok packet, generate an ok packet interrupt to a host system 60 and go to step 160. If the data packet is an error packet, go to step 140.

[0023] Step 140:The counter 46 monitors the number of descriptors 74 in the unavailable state to produce a count value CV. The number of descriptors 74 that change from free to unavailable correspond to the number of data buffers 72 that become inaccessible to the NIC 10. The counter 46 is incremented according to the number of descriptors that change from free to unavailable.

[0024] Step 150:The comparator 42 compares the count value CV of the counter 46 and the threshold value TH of the threshold memory 44. If the CV has reached the TH, go to step 160. If the CV has not reached the TH, return to step

110. Note that the threshold value is programmable.

[0025] Step 160:Because the CV has reached the TH, the NIC 10 issues a match interrupt to the host system 60, and resets the counter 46 to an initial value.

[0026] Step 170:All unavailable descriptors are marked free by the host system.

[0027] When the Receive DMA 40 transfers a data packet into a data buffer 72, the NIC 10 must obtain information about the data buffer 72 from a descriptor 74 corresponding to that data buffer 72. When the information is obtained, the NIC 10 marks the own bit of the corresponding descriptor 74 and its state goes from free to unavailable. Also, please remember that a single data packet may be so long that it requires two or more data buffers 72 to store the data packet. As a result, the number of descriptors that have their state changed from free to unavailable is determined in Step 120.

[0028] In Step 150, Because the CV has reached the TH, it means that the number of unavailable descriptors 74 is at a level that is no longer acceptable for the NIC 10; remember the number of free descriptors 74 corresponds to the number of data buffers 72 the NIC 10 has access. Therefore, the NIC 10 in Step 160 issues a match signal to the host sys-

tem 60. Because the masking circuit 50 is off with regards to the match signal, the match signal sent from the comparator 42 is able to notify the host system 60. The host system 60 as a result clears the data buffers 72 that correspond to the descriptors 74 in an unavailable state. By clearing the data buffers 72, it means that the number of free descriptors returns to the original number. As a result, in Step 160, the NIC 10 also resets the value of the counter 46 to an initial value.

[0029] Please note the following. In the preferred embodiment explained in the specification, the counter tracks the number of unavailable descriptors 74. However, the counter 46 could just as easily track the number of free descriptors 74 remaining. In this case, instead of counting up, the counter 46 counts down. Also, note that even though a masking circuit 50 is depicted in Fig.1, the masking circuit is not necessarily and therefore, should not be taken as a limitation. As stated previously, most host systems 60 would like to be notified as soon as possible when an ok packet is received, and even though the error signal is blocked from notifying the CPU 80 of the host system 60, there are other parts of the NIC 10 or host system that would like to track the number of error

packets received. Therefore, a masking circuit 50 is employed to allow the ok signal to pass through to the CPU 80 but not the error signal. One final note, the comparator 42, threshold memory 44, and counter 46 are not limited to being positioned within the Receive DMA module 40.

[0030] As one can see, the present invention clearly has advantages over the prior art. Because the prior art coalesces events but an error packet event may cause more than one descriptor to be changed, the prior art is suspect to dropping packets, and therefore, negatively effect the host system's management of received data packets from a network. Therefore, the present invention is able to improve the host system's management of received data packets from a network.

[0031] Those skilled in the art will readily observe that numerous modifications and alterations of the device may be made while retaining the teachings of the invention. Accordingly, that above disclosure should be construed as limited only by the metes and bounds of the appended claims.